

데이터베이스시스템 과제 2-1 설명문서

<한양포탈의 '주소록 관리'>

2017030519 컴퓨터소프트웨어학부 홍유진

PART ONE : 로그인

- 화면 출력 코드

```
print("=====LOGIN=====\n")
inputid = input("ID : ")
password = input("password : ")
print("=====LOGIN=====\n")
```

- 실행시 화면

```
PS C:\Users\user\Desktop\dbassignment> python .\tempassign.py
=====LOGIN=====

ID : admin
password : 1111
```

1. admin일 때

: 비밀번호를 상관하지 않고 admin이면 무조건 관리자모드로 실행한다.

2. 포탈 이용자일 때

```
#학생모드
while 1:
    ok=0
    f = open('students.csv','r',encoding = 'utf-8',newline = '')
    rdr = csv.reader(f)
    for line in rdr: #비밀번호와 맞는지 비교하기
        iddel = ''.join(line[0].split()) #학번 공백제거
        if iddel==inputid: # id가 students.csv파일에 있을 때
            passdel = ''.join(line[1].split()) # password공백제거
            if password==passdel: #비밀번호가 같은게 있는지 확인
                ok=1
                break
    f.close()
    if ok==0:
        print("존재하지 않거나 비밀번호가 틀렸습니다.")
        break
```

: students.csv에서 id와 비밀번호가 다르거나 id를 찾지 못했을 때 "존재하지 않거나 비밀번호가 틀렸습니다."라는 메시지를 출력한다.

- 출력 화면

```
PS C:\Users\user\Desktop\dbassignment> python .\tempassign.py
=====LOGIN=====

ID : 2009003125
password : 1111
=====

존재하지 않거나 비밀번호가 틀렸습니다.
```

: 원래의 학번 2009003125의 비밀번호는 125125125이므로 틀렸다는 메시지가 뜬다.

PART TWO : 포탈 관리자 모드

: ID가 admin일 때 실행

- 관리자 모드 명령수행 화면

```
>>>>>>>관리자 모드<<<<<<<<<<
=====어떤 명령을 수행하시겠습니까?=====

a. 새학생 추가
b. 연락처 추가기능
c. 학생 데이터 변경
d. 학생 정보 삭제
e.도메인별 도수분포표 보기
f. 종료
=====
원하는 명령을 입력하시오 : █
```

1. a. 새 학생 추가

- 새로운 학생 '홍길동' 을 추가한다.
 - 2016001234, xxx, 홍길동, male, 6, 1999002345, 1

: 과제시나리오의 이 부분을 수행한다.

- 코드

```
19 elif com == 'a':
20     #새학생 추가기능
21     print("새로운 학생의 정보를 입력하세요 : ")
22     new_student = input()
23     new_student = new_student.split(',')
24     f = open ( 'students.csv','a',encoding = 'utf-8',newline = '')
25     wr = csv.writer(f)
26     wr.writerow(new_student)
27     f.close()
```

: new_stuent에 새로운 학생의 정보 ex) 2016001234, xxx, 홍길동, male, 6, 1999002345, 1
넣고 파일의 모드를 'a'로 하여 students.csv파일 마지막에 학생의 정보를 write한다.

- 코드 실행화면

1) 실행전

tempassign.py	students.csv	한양인 주소록.csv
19	2014019493	,zerozero,임지훈,male,17,2001032003,1
20	2014040404	,donkey,권지현,female,22,2001032053,1
21	2014505050	,hellobaby,권희조,female,44,2001032078,1
22		

2) 실행 후

```

=====어떤 명령을 수행하시겠습니까?=====
a. 새학생 추가
b. 연락처 추가기능
c. 학생 데이터 변경
d. 학생 정보 삭제
e.도메인별 도수분포표 보기
f. 종료
=====
원하는 명령을 입력하시오 : a
새로운 학생의 정보를 입력하세요 :
2016001234, xxx, 홍길동, male, 6, 1999002345, 1

```

tempassign.py	students.csv	한양인 주소록.csv
19	2014019493	,zerozero,임지훈,male,17,2001032003,1
20	2014040404	,donkey,권지현,female,22,2001032053,1
21	2014505050	,hellobaby,권희조,female,44,2001032078,1
22	2016001234, xxx, 홍길동, male, 6, 1999002345, 1	
23		

: students.csv의 22번째줄에 추가된 것을 확인할 수 있다.

2. b. 연락처 추가 기능

연락처 추가 기능을 구현하고, '홍길동'의 연락처를 '한양인 주소록'에 추가한다.

- 2016001234, 01088884444, hong@hanyang.ac.kr

: 과제 시나리오의 이 부분을 수행한다.

- 코드

```

28 elif com == 'b':
29     #연락처 추가기능
30     addcontact = input()
31     addcontact = addcontact.split(',')
32     f = open('한양인 주소록.csv','a',encoding = 'utf-8',newline = '')
33     wr = csv.writer(f)
34     wr.writerow(addcontact)
35     f.close()

```

: a와 비슷하게 한양인 주소록.csv에 새로운 연락처 ex) 2016001234, 01088884444, hong@hanyang.ac.kr를 추가하여 'a'모드로 write한다.

- 코드 실행 화면

1) 코드 실행전

```
tempassign.py students.csv 한양인 주소록.csv
19 2012394929 ,01020374564,sage_wieser@cox.net
20 2012004003 ,01035039847,simona@morasca.com
21 2014040404 ,01085931087,yuki_whobrey@aol.com
22
```

2) 코드 실행 후

```
====어떤 명령을 수행하시겠습니까?=====
a. 새학생 추가
b. 연락처 추가기능
c. 학생 데이터 변경
d. 학생 정보 삭제
e.도메인별 도수분포표 보기
f. 종료
=====
원하는 명령을 입력하시오 : b
2016001234, 01088884444, hong@hanyang.ac.kr
=====
tempassign.py 한양인 주소록.csv x students.csv
19 2012394929 ,01020374564,sage_wieser@cox.net
20 2012004003 ,01035039847,simona@morasca.com
21 2014040404 ,01085931087,yuki_whobrey@aol.com
22 2016001234, 01088884444, hong@hanyang.ac.kr
23
```

: 한양인 주소록.csv에 추가된 것을 확인할 수 있다.

3. c. 학생 데이터 변경

- 연락처 변경 기능을 구현하고, 학생 '권희조'의 이메일을 'kwon@hanyang.ac.kr'로 변경한다.

: 과제시나리오의 이 부분을 수행한다.

- 코드

```

36         elif com == 'c':
37             #연락처 변경기능
38             print("학생이름을 입력하시요\n")
39             stu_name = input()
40             print("변경할 email을 입력하시요\n")
41             new_email = input()
42             f = open ( 'students.csv','r',encoding = 'utf-8',newline = '')
43             rdr = csv.reader(f)
44             for line in rdr: # 이름을 통해 stu_id 찾기
45                 if line[2]==stu_name:
46                     stu_id = line[0]
47                     break
48             f.close()
49             stu_id = ''.join(stu_id.split())
50             #stu_id(학번) 통해서 email 바꿔 temp파일에 적는다.
51             with open ('한양인 주소록.csv','r',encoding = 'utf-8',newline='') as fi:
52                 with open ('temp.csv','w',encoding = 'utf-8',newline='') as fo:
53                     fr = csv.reader(fi, delimiter=',')
54                     fw = csv.writer(fo,delimiter=',')
55                     for row in fr:
56                         row[0]=''.join(row[0].split())
57                         if row[0]==stu_id:
58                             row[2]=new_email
59                             fw.writerow(row)
60                         else :
61                             fw.writerow(row)

```

```

62         #한양인 주소록 비우기
63         with open ('한양인 주소록.csv','w',encoding = 'utf-8',newline='') as f:
64             fw = csv.writer(f,delimiter=',')
65             #temp파일에 있는내용을 한양인 주소록에 복사
66             with open ('한양인 주소록.csv','w',encoding = 'utf-8',newline='') as fo:
67                 with open ('temp.csv','r',encoding = 'utf-8',newline='') as fi:
68                     fr = csv.reader(fi, delimiter=',')
69                     fw = csv.writer(fo,delimiter=',')
70                     for row in fr:
71                         fw.writerow(row)
72             #temp파일 비우기
73             with open ('temp.csv','w',encoding = 'utf-8',newline='') as f:
74                 fw = csv.writer(f,delimiter=',')

```

: 코드 42-48번은 입력 받은 학생이름을 통해 학생의 sid를 찾는 과정이다.

49번은 찾은 sid의 공백을 제거한다.

코드 51번부터 74번은 전체적으로 temp.csv를 활용한다.

코드 51번부터 61번은 한양인 주소록.csv에서 찾은 sid에 일치하는 행을 찾고 변경한 email을 temp.csv에 write한다.

코드 62번부터 64번은 temp.csv에 내용을 한양인 주소록.csv에 다시 write하기위해 한양인 주소록.csv의 내용을 다 지워 빈파일로 만든다.

코드 66번부터 71번은 temp.csv에 내용을 한양인 주소록.csv에 다시 write하는 과정이다.

마지막으로 코드 73부터 74번은 temp.csv를 다시 빈 파일로 만든다.

- 코드 실행 화면

1) 실행전

```

4 2014001303 ,01068220334,calbares@gmail.com
5 2011040404 ,01078554051,donette.foller@cox.net
6 2014505050 ,01062384210,fletcher.flosi@yahoo.com
7 2014019493 ,01056662093,gladys.rim@rim.org
8 2013040051 ,01064989926,gruta@cox.net
9 2009003125 ,01095434668,jbutt@gmail.com

```

: 권희조의 학번은 2014505050으로 fletcher.flosi@yahoo.com의 이메일을 가지고 있다.

2) 코드 실행 후

```

a. 새학생 추가
b. 연락처 추가기능
c. 학생 데이터 변경
d. 학생 정보 삭제
e.도메인별 도수분포표 보기
f. 종료
=====
원하는 명령을 입력하시오 : c
학생이름을 입력하시오

권희조
변경할 email을 입력하시오

kwon@hanyang.ac.kr

```

```

tempassign.py  한양인 주소록.csv x students.csv
6 2014505050,01062384210,kwon@hanyang.ac.kr
7 2014019493,01056662093,gladys.rim@rim.org
8 2013040051,01064989926,gruta@cox.net
9 2009003125,01095434668,jbutt@gmail.com
10 2010004052,01022215104,josephine_darakiv@darakiv.org

```

: 이메일이 변경된 것을 확인할 수 있다.

4. d. 학생 정보 삭제

- 연락처 삭제 기능을 구현하고, 학생 '김다현'의 연락처(번호, 이메일만)를 삭제한다. (학생 정보는 삭제하지 않는다.)

: 과제시나리오의 이 부분을 수행한다.

- 코드

```

75 elif com == 'd':
76     #연락처 삭제
77     print("학생이름을 입력하시요\n")
78     stu_name = input()
79     f = open ('students.csv','r',encoding = 'utf-8',newline = '')
80     rdr = csv.reader(f)
81     #이름을 통해 stu_id찾기
82     for line in rdr:
83         if line[2]==stu_name:
84             stu_id = line[0]
85             break
86     f.close()
87     stu_id = ''.join(stu_id.split())
88     with open ('한양인 주소록.csv','r',encoding = 'utf-8',newline='') as fi:
89         with open ('temp.csv','w',encoding = 'utf-8',newline='')as fo:
90             fr = csv.reader(fi, delimiter=',')
91             fw = csv.writer(fo,delimiter=',')
92             for row in fr:
93                 row[0]=''.join(row[0].split())
94                 if row[0]==stu_id: #위에서 찾은 학번과 같을때
95                     else :|
96                     fw.writerow(row)
97     with open ('한양인 주소록.csv','w',encoding = 'utf-8',newline='') as f:
98         fw = csv.writer(f,delimiter=',')
99     with open ('한양인 주소록.csv','w',encoding = 'utf-8',newline='') as fo:
100         with open ('temp.csv','r',encoding = 'utf-8',newline='')as fi:
101             fr = csv.reader(fi, delimiter=',')
102             fw = csv.writer(fo,delimiter=',')
103             for row in fr:
104                 fw.writerow(row)
105     with open ('temp.csv','w',encoding = 'utf-8',newline='') as f:
106         fw = csv.writer(f,delimiter=',')
107     #말에 과정은 c와 같다.

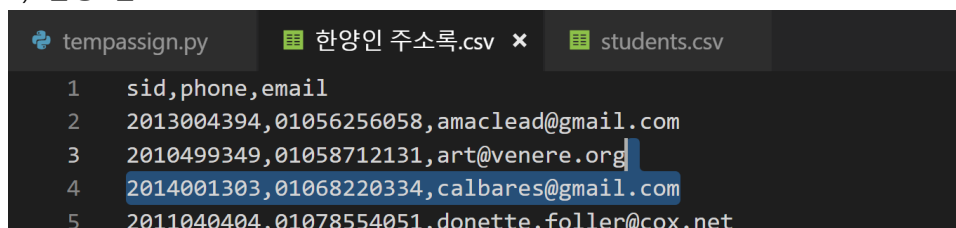
```

: 전체적인 맥락은 c와 비슷하다. c와 같이 학생이름을 통해 students.csv파일에서 sid를 찾고 한양인 주소록에서 찾은 sid를 가진 행을 변경해주어 temp에 적고 한양인 주소록.csv의 내용을 비우고 temp.csv의 내용을 한양인 주소록에 적고 temp.csv의 내용을 비워준다.

c와 다른 것은 94번째 줄처럼 만약 지우려는 student_id이면 temp에 write하지 않음으로써 결국 한양인주소록에서 delete할 수 있다.

- 코드 실행

1) 실행 전



```

1 sid,phone,email
2 2013004394,01056256058,amaclead@gmail.com
3 2010499349,01058712131,art@venere.org
4 2014001303,01068220334,calbares@gmail.com
5 2011040404,01078554051,donette.foller@cox.net

```

: 김다현의 sid는 2014001303이다.

2) 실행 후

```
a. 새학생 추가
b. 연락처 추가기능
c. 학생 데이터 변경
d. 학생 정보 삭제
e.도메인별 도수분포표 보기
f. 종료
=====
원하는 명령을 입력하시오 : d
학생이름을 입력하시오

김다현
```

```
tempassign.py  한양인 주소록.csv x students.csv
1  sid,phone,email
2  2013004394,01056256058,amaclead@gmail.com
3  2010499349,01058712131,art@venere.org
4  2014001303,,
5  2011040404,01078554051,donette.foller@cox.net
```

: phone과 email이 지워진 것을 확인 할 수 있다.

5. e. 도메인별 도수분포표 보기

- 관리자 모드에서:
 - '한양인 주소록' 파일의 email 열을 'local_part' 열과 'domain_name' 열로 분리하여 저장한다.
 - @ 앞 부분이 localpart, 뒤 부분이 domain name 이다.
 - hong@hanyang.ac.kr 이라면 localpart 는 hong, domain 은 hanyang.ac.kr 이다.
 - 도메인 별로 도수 분포표(빈도표)를 볼 수 있는 기능을 구현한다.

Step 3의 이 부분을 수행하였다.

- 코드


```

111 elif com == 'e':
112     #도메인별 도수분포표
113     f = open ( '한양인 주소록.csv','r',encoding = 'utf-8',newline = '')
114     rdr = csv.reader(f)
115     #이름을 통해 stu_id찾기
116     finddomain=[]#locale part 와 domain 구분
117     count=0
118     for line in rdr:
119         if count!=0:
120             finddomain.append(line[2].split('@'))
121             count=count+1
122     f.close()
123     domain = {}
124     for row in finddomain:
125         #print(row)
126         n=0
127         for col in row:
128             if n==1:
129                 domain[col]=0
130             n=n+1
131     for row in finddomain:
132         n=0
133         for col in row:
134             if n==1:
135                 domain[col]=domain[col]+1
136             n=n+1
137     #domain에는 각 domain별 분포도가 나타나있다.
138     d=pd.Series(domain)
139     print(d) #도수분포표 보기

```

: 코드 118-122는 finddomain이라는 list에 @를 구분에서 인자를 추가한다. Count가 있는 것은 count가 0일때는 열의 이름(sid,phone,email)을 구분해 finddomain에 넣어주어 이를 방지하기 위함이다.

코드 123- 130은 domain이라는 dictionary에 finddomain에서 찾은 도메인 값을 넣어준다. 이때 value값은 0으로 초기화 한다.

코드 131-136은 각 도메인의 개수를 세서 value값을 바꿔준다.

코드 138은 import pandas as pd를 통해 Series를 불러 domain을 도수분포표로 볼 수 있도록 d에 저장한후 139번에서 d를 출력한다.

- 코드 실행 화면

1) 실행 후

: 이렇게 도수분포표를 확인할 수 있다.

: while문을 종료한다.

```
=====LOGIN=====

ID : 2009003125
password : 125125125
=====

=====어떤 명령을 수행하시겠습니까?=====

a. 메일보내기 기능
b. 개인주소록 파일 추가
c. 개인주소록 업데이트
d. 개인주소록 연락처 삭제
e. 종료

=====
원하는 명령을 입력하십시오 :
```

1. b. 개인주소록 파일 추가

- THE LOW-COST, HIGH-QUALITY SOLUTION**

과제의 이부분을 수행한다.

- 코드

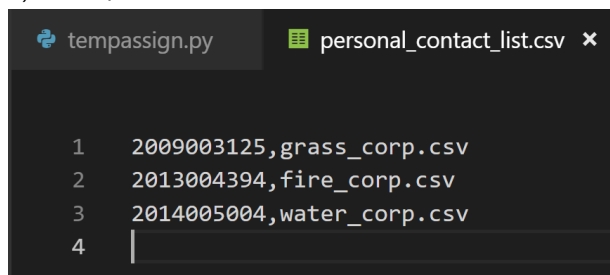
```
187 elif com_stu == 'b' :
188     #학번으로 로그인한다. id를 찾는다.
189     print("추가할 파일을 입력하시오 : ")
190     addfilename = input()
191     c = inputid+'_'+addfilename
192     c = c.split(',')
193     f = open ( 'personal_contact_list.csv','a',encoding = 'utf-8',newline = '')
194     wr = csv.writer(f)
195     wr.writerow(c)
196     f.close()
```

: personal_contact.csv를 새로 만든다. 이 파일은 학생 id와 개인주소록 파일명을 열로 가지고 있다.

- 코드 수행

: 코드 수행을 위해 정남아 , 윤인욱, 장두호로 각각 login하여 개인주소록을 생성한다.

1) 실행후



```
tempassign.py personal_contact_list.csv x
1 2009003125,grass_corp.csv
2 2013004394,fire_corp.csv
3 2014005004,water_corp.csv
4 |
```

2. c. 개인주소록 업데이트

- '정남아', '윤인욱', '장두호'의 개인주소록에 business_card_updated.zip 을 각각 반영
 - '전화번호'가 같으면 동일인으로 취급. 다른 정보를 업데이트

과제 시나리오의 이 부분을 수행한다.

<ver 1 : 원래파일에 업데이트된 부분을 추가할 때, 즉 원래파일에 없던 주소록이 업데이트 파일에 생겼을 때, 원래파일에 있던주소록에 업데이트파일엔 없어졌을 때>

(업데이트부분이 잘 이해가 안되 두가지 버전으로 만들어보았습니다. python파일은 ver1은 주석처리하고 ver2로 보냈습니다.)

- 코드

```
209 elif com_stu == 'c' :
210     #정보를 업데이트 전화번호가 다른것만
211     #id를 통해 개인주소록 파일에 접근해 업데이트하자
212     f = open ( 'personal_contact_list.csv','r',encoding = 'utf-8',newline = '')
213     rdr = csv.reader(f)
214     for line in rdr: # updatefile에 sid를 통한 개인주소록을 받아온다.
215         if line[0] ==inputid:
216             updatefilename = line[1]
217             break
218     f.close()
219     #temp 에 업데이트된 파일을 적음
220     u = 'business_cards_updated/'+updatefilename
221     with open (u,'r',encoding = 'utf-8',newline='') as fi:
222         with open ('temp.csv','w',encoding = 'utf-8',newline='')as fo:
223             fr = csv.reader(fi, delimiter=',')
224             fw = csv.writer(fo,delimiter=',')
225             for row in fr:
226                 fw.writerow(row)
227     addnum=[]
228     with open ('임시주소록 업데이트.csv','w',encoding = 'utf-8',newline='')as fo:
229         with open ('temp.csv','r',encoding = 'utf-8',newline='')as tempi:
230             fw = csv.writer(fo,delimiter=',')
231             tempr = csv.reader(tempi,delimiter=',')
232             for row in tempr:
233                 numberexist =0
234                 f = open ( updatefilename,'r',encoding = 'utf-8',newline = '')
235                 fr = csv.reader(f)
236                 for col in fr:
237                     if(row[1]==col[1]): #업데이트 된 전화번호와 원래 전화번호가 같을 때
238                         fw.writerow(row)
239                         numberexist=1
240                         break
241                 f.close()
242                 if numberexist==0:
243                     print("전화번호 없음 : "+row[1]+col[1])
244                     addnum.append(row)# 업데이트 된 전화번호가 원래 파일에 없을때 리스트에 추가
245
246     #원래파일 초기화
247     with open (updatefilename,'w',encoding = 'utf-8',newline='') as f:
248         fw = csv.writer(f,delimiter=',')
249     #임시주소록 업데이트.csv파일을 원래파일에 적기
250     with open (updatefilename,'w',encoding = 'utf-8',newline='') as fo:
251         with open ('임시주소록 업데이트.csv','r',encoding = 'utf-8',newline='')as fi:
252             fr = csv.reader(fi, delimiter=',')
253             fw = csv.writer(fo,delimiter=',')
254             for row in fr:
255                 fw.writerow(row)
256
257     #리스트에 있는 내용을 원래파일에 추가로 적기
258     with open (updatefilename,'a',encoding = 'utf-8',newline='') as fo:
259         fw = csv.writer(fo,delimiter=',')
260         for row in addnum:
261             fw.writerow(row)
262     print(addnum)
263     #temp.csv초기화
264     with open ('temp.csv','w',encoding = 'utf-8',newline='') as f:
265         fw = csv.writer(f,delimiter=',')
266     #임시주소록 업데이트.csv초기화
267     with open ('임시주소록 업데이트.csv','w',encoding = 'utf-8',newline='') as f:
268         fw = csv.writer(f,delimiter=',')
269
```

: 전체 수행 내용은 update파일에 대한 내용을 temp.csv에 적은 후 원래 파일에 다시 적는다. 코

드 200-208은 업데이트 파일에 대한 내용을 temp.csv에 적는다. 이때 u라는 변수를 통해 business_cards_updated 폴더 안에 있는 파일이라는 경로를 정해준다.

코드 227-224는 temp.csv의 파일과 원래파일에서 전화번호가 같으면 임시주소록 업데이트.csv라는 새로운 파일에 temp.csv의 행을 적는다. 그 이유는 번호가 같아도 다른 정보가 바뀔 수 있기 때문에 새로 업데이트된 행을 적는다. 그리고 전화번호가 없는 행들은 addnum이라는 리스트에 따로 저장해준다.

코드 247-248은 원래 파일이 내용을 비운다. 새로운 내용을 다시 적기 위함이다.

코드 249-255는 임시주소록 업데이트.csv에 적힌내용을 원래파일에 적는다.

코드 258-262에는 addnum리스트에 있는 행을 원래파일에 적는다.

마지막으로 temp.csv와 임시주소록 업데이트.csv파일의 내용을 비운다.

- 코드 실행 화면

1) 실행 전

번호가 없었던것도 예제로 보기 위하여 정남아로 로그인한다.

```
tempassign.py  Grass_corp.csv C:\...\dbassignment  Grass_corp.csv .\
1  이상해씨, 01023140011, bulbasaur@grass.poke, 사장
2  치코리타, 01051522001, chikorita@grass.poke, 부장
3  주리비안, 01064950101, snivy@grass.poke, 과장
4  나무지기, 01032540033, sceptile@grass.poke, 사원z
```

: 업데이트되기전 원래의 파일의 내용이다.

```
tempassign.py  Grass_corp.csv C:\...\dbassignment  Grass_corp.csv .\
1  이상해풀, 01023140011, ivysaur@grass.poke, 이사
2  치코리타, 01051522001, meganimum@grass.poke, 사장
3  리피아, 01061344185, leafeon@grass.poke, 부장
4  주리비안, 01064950101, snivy@grass.poke, 과장
5  나무지기, 01032540033, sceptile@grass.poke, 사원
6
```

:업데이트해야할 파일의 내용이다. (business_card_updated.zip의 파일)

2) 실행후

```
====어떤 명령을 수행하시겠습니까?=====
a. 메일보내기 기능
b. 개인주소록 파일 추가
c. 개인주소록 업데이트
d. 개인주소록 연락처 삭제
e. 종료
=====
원하는 명령을 입력하시오 :
c
전화번호 없음 : 01061344185 01032540033
[['리피아', '01061344185', 'leafeon@grass.poke', '부장']]
어떤 명령을 수행하시겠습니까?
```

: 전화번호가 없는 행을 확인하기 위해 잠시 출력했다. '리피아'의 번호가 없는 것을 확인했다.

```
tempassign.py  Grass_corp.csv C:\...\dbassignment  Grass_corp.csv \ x
1  이상해풀, 01023140011, ivysaur@grass.poke, 이사
2  치코리타, 01051522001, meganium@grass.poke, 사장
3  리피아, 01061344185, leafeon@grass.poke, 부장
4  주리비안, 01064950101, snivy@grass.poke, 과장
5  나무지기, 01032540033, sceptile@grass.poke, 사원
6
```

:업데이트해야 할 파일의 내용이다. (business_card_updated.zip의 파일)

```
tempassign.py  Grass_corp.csv C:\...\dbassignment x  Grass_corp.csv \  Wa
1  이상해풀, 01023140011, ivysaur@grass.poke, 이사
2  치코리타, 01051522001, meganium@grass.poke, 사장
3  주리비안, 01064950101, snivy@grass.poke, 과장
4  나무지기, 01032540033, sceptile@grass.poke, 사원
5  리피아, 01061344185, leafeon@grass.poke, 부장
6
```

:실행 후 바뀐 파일의 내용이다. 이상해씨가 이상해풀로 바뀌고 리피아의 연락처가 추가된 것을 확인할 수 있다.

<ver2> 업데이트파일로 원래파일을 아이에 바꿀 때

- 코드

```

205 elif com_stu == 'c' :
206     #정보를 업데이트 전화번호가 다른것만
207     #id를 통해 개인주소록 파일에 접근해 업데이트하자
208     f = open ( 'personal_contact_list.csv','r',encoding = 'utf-8',newline = '')
209     rdr = csv.reader(f)
210     for line in rdr: # updatefile에 sid를 통한 개인주소록을 받아온다.
211         if line[0] ==inputid:
212             updatefilename = line[1]
213             break
214     f.close()
215     #업데이트 파일로 원래파일을 변경할때
216     #temp 에 업데이트된 파일을 적음
217     u = 'business_cards_updated/'+updatefilename
218     with open (u,'r',encoding = 'utf-8',newline='') as fi:
219         with open ('temp.csv','w',encoding = 'utf-8',newline='')as fo:
220             fr = csv.reader(fi, delimiter=',')
221             fw = csv.writer(fo,delimiter=',')
222             for row in fr:
223                 fw.writerow(row)
224     #원래파일 초기화
225     with open (updatefilename,'w',encoding = 'utf-8',newline='') as f:
226         fw = csv.writer(f,delimiter=',')
227
228     #temp에 있는 내용을 원래파일에 적는다.
229     with open (updatefilename,'w',encoding = 'utf-8',newline='') as fo:
230         with open ('temp.csv','r',encoding = 'utf-8',newline='')as fi:
231             fr = csv.reader(fi, delimiter=',')
232             fw = csv.writer(fo,delimiter=',')
233             for row in fr:
234                 fw.writerow(row)
235     #temp.csv초기화
236     with open ('temp.csv','w',encoding = 'utf-8',newline='') as f:
237         fw = csv.writer(f,delimiter=',')

```

: 업데이트파일을 temp.csv에 옮겨적은후 temp.csv에 적은것을 다시 원래파일에 적는다.
 이때 temp파일은 초기화시킨다.

- 코드실행 후

	2017030519_hw2-1_코드.py	students.csv	Grass_corp.csv business_cards_updated
1	이상해풀, 01023140011, ivysaur@grass.poke, 이사		
2	치코리타, 01051522001, meganium@grass.poke, 사장		
3	리피아, 01061344185, leafeon@grass.poke, 부장		
4	주리비안, 01064950101, snivy@grass.poke, 과장		
5	나무지기, 01032540033, sceptile@grass.poke, 사원		
6			

: business_Cards_updated에 있는 파일

```
2017030519_hw2-1_코드.py students.csv Grass_corp.csv business_cards_updated Grass_corp.csv .\ x
1 이상해풀, 01023140011, ivysaur@grass.poke, 이사
2 치코리타, 01051522001, meganium@grass.poke, 사장
3 리피아, 01061344185, leafeon@grass.poke, 부장
4 주리비안, 01064950101, snivy@grass.poke, 과장
5 나무지기, 01032540033, sceptile@grass.poke, 사원
6
```

: 원래 파일이 update파일처럼 변경된 것을 확인할 수 있다.

3. d. 개인주소록 연락처 삭제

- '정남아', '윤인욱', '장두호'의 개인주소록에서 임의의 연락처 1 건을 삭제

과제파일의 이부분을 수행한다.

- 코드

```
222 elif com_stu == 'd':
223     #input id와 개인주소록을 연동해야함
224     f = open ('personal_contact_list.csv','r',encoding = 'utf-8',newline = '')
225     rdr = csv.reader(f)
226     for line in rdr: # deletefile에 sid를 통한 개인주소록을 받아온다.
227         if line[0] ==inputid:
228             deletefile = line[1]
229             break
230     f.close()
231     #temp에 옮겨적음
232     with open (deletefile,'r',encoding = 'utf-8',newline='') as fi:
233         with open ('temp.csv','w',encoding = 'utf-8',newline='')as fo:
234             fr = csv.reader(fi, delimiter=',')
235             fw = csv.writer(fo,delimiter=',')
236             for row in fr:
237                 fw.writerow(row)
238     #원래 파일 초기화
239     with open (deletefile,'w',encoding = 'utf-8',newline='') as f:
240         fw = csv.writer(f,delimiter=',')
241     #temp에 옮긴것을 통해 행갯수 찾기
242     rowcnt=0
```



```

243     with open ('temp.csv','r',encoding = 'utf-8',newline='')as fi:
244         fr = csv.reader(fi, delimiter=',')
245         for row in fr:
246             rowcnt=rowcnt+1 #행갯수세기
247     random = randint(0,rowcnt) # 지울 임의의 난수 생성
248     #temp에 옮긴것을 원래파일에 다시 적기
249     i=0
250     with open (deletefile,'w',encoding = 'utf-8',newline='') as fo:
251         with open ('temp.csv','r',encoding = 'utf-8',newline='')as fi:
252             fr = csv.reader(fi, delimiter=',')
253             fw = csv.writer(fo,delimiter=',')
254             for row in fr:
255                 if(i!=random):
256                     fw.writerow(row)
257                 i=i+1
258     #temp 파일 초기화
259     with open ('temp.csv','w',encoding = 'utf-8',newline='') as f:
260         fw = csv.writer(f,delimiter=',')

```

: 코드 224 - 230 은 학번을 통해 personal_contact_list.csv에 있는 학번의 개인주소록을 찾아낸다. 전체적인 코드내용은 원래 파일 내용을 temp.csv에 복사 한 후 수정 후 다시 원래 파일에 적는것이다. 코드 232-240은 deletefile(개인주소록) 내용을 temp.csv에 옮겨 적은 후 원래 파일을 초기화 한다.

코드 242-248은 temp.csv의 행의 개수를 센다. 개수를 세는 이유는 다음코드에 있다.

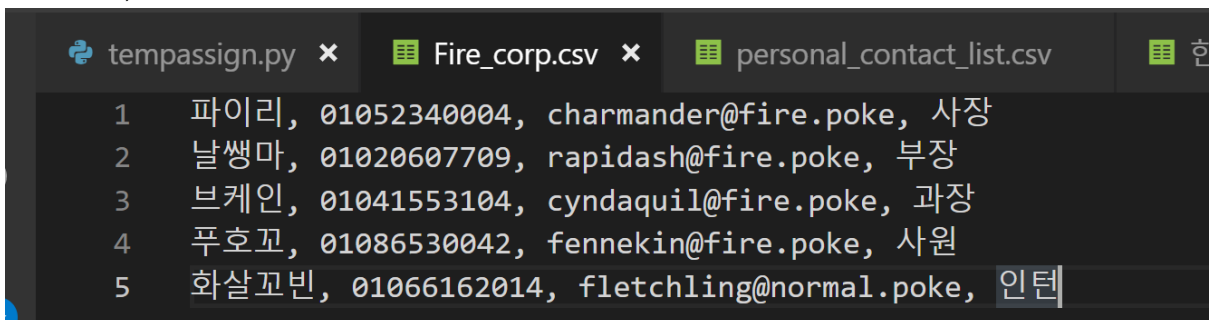
코드 247은 행 개수 사이의 random수를 생성한다. 이 수를 통해 이 행을 지울 것이다.

코드 225-257은 random으로 생성된 수의 행을 제외한 나머지 행을 원래의 파일에 적는다. 그 후 temp파일을 초기화 한다.

- 코드 실행 화면

윤인욱으로 로그인하여 fire_corp.csv를 수정한다고 하자

1) 실행전



```

1   파이리, 01052340004, charmander@fire.poke, 사장
2   날쌔마, 01020607709, rapidash@fire.poke, 부장
3   브케인, 01041553104, cyndaquil@fire.poke, 과장
4   푸호꼬, 01086530042, fennekin@fire.poke, 사원
5   화살꼬빈, 01066162014, fletchling@normal.poke, 인턴

```

2) 실행 후

- a. 메일보내기 기능
- b. 개인주소록 파일 추가
- c. 개인주소록 업데이트
- d. 개인주소록 연락처 삭제
- e. 종료

=====

원하는 명령을 입력하시오 :

d

0

(0은 생성된 난수로 실행을 위해 임의로 출력하였다.)

```
tempassign.py  Fire_corp.csv ×  personal_contact_list.csv
1  날쌔마, 01020607709, rapidash@fire.poke, 부장
2  브케인, 01041553104, cyndaquil@fire.poke, 과장
3  푸호꼬, 01086530042, fennekin@fire.poke, 사원
4  화살꼬빈, 01066162014, fletchling@normal.poke, 인턴
5  |
```

: 0번째 행이 삭제된 것을 확인할 수 있다.

4. a. 메일 보내기 기능

– 메일 보내기 기능에서 ‘한양인 주소록’과 자신의 주소록을 같이 보여줘야 한다.
과제시나리오의 이 부분을 수행한다.

- 코드

```
166 elif com_stu == 'a':
167     #####personal contact 있을 때 어떻게 되는지 확인하기
168
169     #메일보내기 한양인주소록과 개인주소록을 모두 출력
170     print("메일을 보냅니다. 주소록을 확인하세요")
171     print("=====주소록 =====")
172     #input id와 개인주소록을 연동해야함
173     f = open ( 'personal_contact_list.csv','r',encoding = 'utf-8',newline = '')
174     rdr = csv.reader(f)
175     personal=''
176     for line in rdr: # personal에 sid를 통한 개인주소록을 받아온다.
177         if line[0] ==inputid:
178             personal = line[1]
179             break
180     f.close()
181
```

```

182 #한양인 주소록 출력
183 print(">>>>>>>한양인 주소록 <<<<<<<<<<<")
184 f = open ('한양인 주소록.csv','r',encoding = 'utf-8',newline = '')
185 rdr = csv.reader(f)
186 for line in rdr:
187     print(line)
188 f.close()
189 print("=====")
190 if personal != '':#개인주소록이 있을 때
191     print(">>>>>>>>>개인주소록<<<<<<<<<<<") #개인주소록 출력
192     f = open ( personal,'r',encoding = 'utf-8',newline = '')
193     rdr = csv.reader(f)
194     for line in rdr:
195         print(line)
196     f.close()
197     print("=====")

```

: 코드 173 - 180은 ID를 통해 개인주소록 이름을 찾는다. 만약 개인주소록이 없다면 개인주소록을 저장하는 personal변수는 "이다.

코드 182-188은 한양인 주소록을 출력한다.

코드 190-197은 만약 personal이 "이 아닐 때 즉, 개인주소록이 있을 때 개인주소록을 출력한다.

- 코드 실행 화면

```
>>>>>>>>개인주소록<<<<<<<<<<<<<<
['파이리', '01052340004', 'charmander@fire.poke', '사장']
['날쌘마', '01020607709', 'rapidash@fire.poke', '부장']
['브케인', '01041553104', 'cyndaquil@fire.poke', '과장']
['푸호꼬', '01086530042', 'fennekin@fire.poke', '사원']
['화살꼬빈', '01066162014', 'fletchling@normal.poke', '인턴']
=====
```

메일을 보냅니다. 주소록을 확인하세요

[illegible]

=====

```
['파이리', '01052340004', 'charmander@fire.poke', '사장']
```

```
['브케인', '01041553104', 'cyndaquil@fire.poke', '과장']
```

```
['화살꼬빈', '01066162014', 'fletchling@normal.poke', '01066162014']
```

1. *Journal of the American Medical Association*, 1997; 277: 1001-1005.